

APPENDIX 1: SOURCE CODE

TITLE: ACCEPTING USER CONTROL

APPLICANTS: NEIL GELFOND, DAMIAN HOWARD, JOSEPH
KILLOUGH, ANDREW OLCOTT, PETER C. SANTORO,
JAMES SHANLEY AND LEE ZAMIR

Charles Hieken, Reg. No. 18,411
Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110-2804
Tel.: (617) 542-5070
Fax: (617) 542-8906

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EV 331 654 519 US

April 5, 2004
Date of Deposit

```

*****
* Name   : Snap
* Author : Neil Gelfond
* Notice : Copyright (c) 2002 Bose Corporation
*         : All Rights Reserved
* Date   : 08/18/03
* Version : 1.4
* Notes  :
*
*****

```

'This program runs on the PIC Mother Board
'and send actual ascii codes

```

'cmcon      = 7      'disable comparitors
OPTION_REG.7 = 0      'enable weak pull-ups
adcon1      = $0e

```

```

switch      var porta.0

```

```

tx          var portb.2
tx2         var portb.7
mode        var word

```

```

'mode = 84      '9600 non-inverted
'mode = 16468   '9600 inverted
mode = 188      '4800 non-inverted
'mode = 16572   '4800 inverted

```

```

define adc_bits 8
define adc_clock 3
define adc_sampleus 50

```

```

value       var byte
res_value   var byte
zone        var byte
prev_zone   var byte
current_zone var byte
zone_buff   var byte[3]

```

```

input switch

```

```

'define hser_rcsta 90h
'define hser_txsta 20h
'define hser_baud 4800
'define hser_spbrg 12

```

pinA var porta.3
pinB var porta.4

pinC var portb.0
pinD var portb.1

input pinA
input pinB
input pinC
input pinD

counterA var word
counterB var word
counterC var word
counterD var word

bufcount var word
ticcount var byte

tempstateA var bit
tempstateB var bit
tempstateC var bit
tempstateD var bit

prevstateA var bit
prevstateB var bit
prevstateC var bit
prevstateD var bit

stateA var bit
stateB var bit
stateC var bit
stateD var bit

flagA var bit
flagAP var bit
flagB var bit
flagBP var bit
flagC var bit
flagCP var bit
flagD var bit
flagDP var bit

limit con 3
PlusMinus var byte

```
old_PlusMinus var byte
buffer        var byte[3]
dir           var byte
pulsedata     var byte
```

```
ticcount  = 0
dir       = 0
pulsedata = 0
```

```
flagA  = 0
flagAP = 0
```

```
flagB  = 0
flagBP = 0
```

```
flagC  = 0
flagCP = 0
```

```
flagD  = 0
flagDP = 0
```

```
stateA = pinA
stateB  = pinB
stateC  = pinC
stateD  = pinD
```

```
prevstateA = pinA
prevstateB = pinB
prevstateC = pinC
prevstateD = pinD
```

```
tempstateA = pinA
tempstateB = pinB
tempstateC = pinC
tempstateD = pinD
```

```
counterA = limit
counterB = limit
counterC = limit
counterD = limit
*****ZoneInit*****
```

```
zone_buff(0) = $ff
zone_buff(1) = $ff
zone_buff(2) = $ff
zone         = 0
```

```
prev_zone = 0
current_zone = 0
```

```
ADCIN 2,value
```

```
value = value
res_value = value
```

```
*****bufinit*****
```

```
bufinit:
```

```
buffer(0) = $ff
```

```
buffer(1) = $ff
```

```
buffer(2) = $ff
```

```
bufcount = 0
```

```
*****main*****
```

```
main:
```

```
'gosub initialize
```

```
gosub Rotory
```

```
gosub Direction
```

```
gosub BufStuff
```

```
gosub Display
```

```
*****
```

```
gosub resvalue
```

```
gosub Look
```

```
gosub a2dbuffer
```

```
gosub Press_Test
```

```
gosub Release_test
```

```
goto main
```

```
*****initialize*****
```

```
initialize:
```

```
stateA = pinA
```

```
stateB = pinB
```

```
stateC = pinC
```

```

stateD    = pinD
prevstateA = pinA
prevstateB = pinB
prevstateC = pinC
prevstateD = pinD

return
"Rotary"
Rotory:

gosub CheckPinA

gosub CheckPinB

gosub CheckPinC

gosub CheckPinD

if (flagA = 1) or (flagB = 1) then
    pulsedata.0 = prevstateA
    pulsedata.1 = prevstateB
    pulsedata.2 = stateA
    pulsedata.3 = stateB
    flagA = 0
    flagB = 0
    bufcount = 0
else
    if (flagC = 1) or (flagD = 1) then
        pulsedata.0 = prevstateC
        pulsedata.1 = prevstateD
        pulsedata.2 = stateC
        pulsedata.3 = stateD
        flagC = 0
        flagD = 0
        bufcount = 0
    endif
endif

exit_rotory:

return
"CheckPinA"
CheckPinA:

if counterA = limit then
    goto statecheckA

```

```

else
  counterA = counterA + 1
  goto countercheckA
endif

```

```

countercheckA:
if counterA = limit then
  if tempstateA <> pinA then
    goto resetcounterA
  else
    prevstateA = stateA
    stateA = tempstateA
    flagA = 1
    flagAP = 1
    goto exitA
  endif
else
  goto tempstatecheckA
endif

```

```

tempstatecheckA:
if tempstateA <> pinA then
  goto resetcounterA
else
  goto exitA
endif

```

```

statecheckA:
if stateA <> pinA then
  goto resetcounterA
else
  goto exitA
endif

```

```

resetcounterA:
flagA = 0
counterA = 0
tempstateA = pinA

```

```

exitA:
return
"CheckPinB"
CheckPinB:

```

```

if counterB = limit then
  goto statecheckB

```

```

else
    counterB = counterB + 1
    goto countercheckB
endif

countercheckB:
if counterB = limit then
    if tempstateB <> pinB then
        goto resetcounterB
    else
        prevstateB = stateB
        stateB = tempstateB
        flagB = 1
        flagBP = 1
        goto exitB
    endif
else
    goto tempstatecheckB
endif

tempstatecheckB:
if tempstateB <> pinB then
    goto resetcounterB
else
    goto exitB
endif

statecheckB:
if stateB <> pinB then
    goto resetcounterB
else
    goto exitB
endif

resetcounterB:
flagB = 0
counterB = 0
tempstateB = pinB

exitB:
return
*****CheckPinC*****
CheckPinC:

if counterC = limit then
    goto statecheckC

```



```

else
  counterC = counterC +1
  goto countercheckC
endif

countercheckC:
if counterC = limit then
  if tempstateC <> pinC then
    goto resetcounterC
  else
    prevstateC = stateC
    stateC = tempstateC
    flagC = 1
    flagCP = 1
    goto exitC
  endif
else
  goto tempstatecheckC
endif

tempstatecheckC:
if tempstateC <> pinC then
  goto resetcounterC
else
  goto exitC
endif

statecheckC:
if stateC <> pinC then
  goto resetcounterC
else
  goto exitC
endif

resetcounterC:
flagC = 0
counterC = 0
tempstateC = pinC

exitC:
return
""""""""""CheckPinD""""""""""
CheckPinD:

if counterD = limit then
  goto statecheckD

```

```

else
  counterD = counterD + 1
  goto countercheckD
endif

```

```

countercheckD:
if counterD = limit then
  if tempstateD <> pinD then
    goto resetcounterD
  else
    prevstateD = stateD
    stateD = tempstateD
    flagD = 1
    flagDP = 1
    goto exitD
  endif
else
  goto tempstatecheckD
endif

```

```

tempstatecheckD:
if tempstateD <> pinD then
  goto resetcounterD
else
  goto exitD
endif

```

```

statecheckD:
if stateD <> pinD then
  goto resetcounterD
else
  goto exitD
endif

```

```

resetcounterD:
flagD = 0
counterD = 0
tempstateD = pinD

```

```

exitD:
return
"*****Buffer Stuff*****"
BufStuff:

```

```

if (buffer(1) = $ff) or (buffer(2) = $ff) then
  "hserout [hex2 buffer(0)," ",hex2 buffer(1)," ",hex2 buffer(2),13,10]

```

```

goto shift
else
dir.0 = buffer(0)
dir.1 = buffer(1)
dir.2 = buffer(2)
endif

gosub ClockCounterClock

shift:
buffer(2) = buffer(1)
buffer(1) = buffer(0)
'hserout [hex2 buffer(0)," ",hex2 buffer(1)," ",hex2 buffer(2),13,10]

return

*****"Direction"*****
Direction:

select case pulsedata

case 0,5,10,15
buffer(0) = $ff

case 1,7,8,14
buffer(0) = 1

case 2,4,11,13
buffer(0) = 0

case 3,6,9,12
buffer(0) = $ff

end select

return
*****"ClockCounterClock"*****
ClockCounterClock:

select case dir

case 0,1,2,4
PlusMinus = 45 '(- counter-clock-wise)

case 3,5,6,7
PlusMinus = 43 '(+ clock-wise)

```

end select

return

*****Display*****

Display:

if (flagAP = 1) and (flagBP = 1) then

if old_PlusMinus <> PlusMinus then

'ticcount = 0

old_PlusMinus = PlusMinus

endif

'ticcount = ticcount + 1

if PlusMinus = 45 then

'high left

ticcount = ticcount - 1

if ticcount = 0 or ticcount = 255 then

ticcount = 30

'ticcount = 16

endif

serout2 tx,mode,[\$24,\$4c,\$2b,\$2a]

'serout2 tx2,mode,[\$24,\$4c,\$2b,\$2a]

'serout2 tx,mode,[dec ticcount,13,10] '[PlusMinus,13,10]

'hserout [dec ticcount,13,10] '[PlusMinus,13,10]

'pauseus 10000

'low left

else

'high right

ticcount = ticcount + 1

if ticcount = 31 then

'if ticcount = 17 then

ticcount = 1

endif

serout2 tx,mode,[\$24,\$4c,\$2d,\$2a]

'serout2 tx2,mode,[\$24,\$4c,\$2d,\$2a]

'serout2 tx,mode,[dec ticcount,13,10] '[PlusMinus,13,10]

'hserout [dec ticcount,13,10] '[PlusMinus,13,10]

'pauseus 10000

'low right

endif

gosub initialize

flagAP = 0

flagBP = 0

endif

if (flagCP = 1) and (flagDP = 1) then

if old_PlusMinus <> PlusMinus then

'ticcount = 0

old_PlusMinus = PlusMinus

endif

'ticcount = ticcount + 1

if PlusMinus = 45 then

'high left

ticcount = ticcount - 1

if ticcount = 0 or ticcount = 255 then

ticcount = 30

'ticcount = 16

endif

serout2 tx,mode,[\$24,\$52,\$2b,\$2a].

'serout2 tx2,mode,[\$24,\$52,\$2b,\$2a]

'serout2 tx,mode,[dec ticcount,13,10] '[PlusMinus,13,10]

'hserout [dec ticcount,13,10] '[PlusMinus,13,10]

'pauseus 10000

'low left

else

'high right

ticcount = ticcount + 1

if ticcount = 31 then

'if ticcount = 17 then

ticcount = 1

endif

serout2 tx,mode,[\$24,\$52,\$2d,\$2a]

'serout2 tx2,mode,[\$24,\$52,\$2d,\$2a]

'serout2 tx,mode,[dec ticcount,13,10] '[PlusMinus,13,10]

'hserout [dec ticcount,13,10] '[PlusMinus,13,10]

'pauseus 10000

'low right

endif

gosub initialize

flagCP = 0

flagDP = 0

endif

Return

*****Resvalue*****

Resvalue:

ADCIN 0,value

if res_value <> value then

pause 50

res_value = value

goto resvalue

endif

res_value = value

exit_resval:

return

*****Look up*****

Look:

select case res_value

'serout2 tx,mode,["Res Value ",dec res_value,13,10]

case 104,105,106 'top left ,232

zone_buff(0) = \$31

'zone = 1

case 92,93,94,95 'left knob

zone_buff(0) = \$36

'zone = 6

case 178,179,180,181 'bottom left

zone_buff(0) = \$32

'zone = 2

case 117,118,119,120,121 'Top Center

zone = \$35

'zone = 5

case 97,98,99,100,101 'Bottom Center

zone = \$38

'zone = 8

```

case 210,211,212,213,214      'top right
    zone_buff(0) = $33
    'zone = 3

case 129,130,131      'Right knob 157,118,162,106,166,165,117
    zone_buff(0) = $37
    'zone = 7

case 141,142,143,144      'bottom right
    zone_buff(0) = $34
    'zone = 4

'case is < 20
' zone = 0

case else
    zone_buff(0) = 0

end select

exit_lookup:

return
"*****A2D Buffer*****"
a2dbuffer:

if (zone_buff(1) = $ff) or (zone_buff(2) = $ff) then
    goto buff_shift
endif

        "*****compare buffers*****"
if zone_buff(0) = zone_buff(1) then
    zone = zone_buff(0)
    goto clear_buff
endif

if zone_buff(0) = zone_buff(2) then
    zone = zone_buff(0)
    goto clear_buff
endif

if zone_buff(1) = zone_buff(2) then
    zone = zone_buff(1)
    goto clear_buff
endif

```

```

        "*****clear buffers*****"
clear_buff:
zone_buff(0) = $ff
zone_buff(1) = $ff
zone_buff(2) = $ff
pauseus 100
goto exit_a2dbuffer

buff_shift:
zone_buff(2) = zone_buff(1)
zone_buff(1) = zone_buff(0)

exit_a2dbuffer:

return
*****Press Test*****
Press_Test:

if zone <> current_zone then
if zone <> 0 then
current_zone = zone
gosub press
endif
endif

exit_press_test:
return
*****Release Test*****
Release_Test:

if zone <> current_zone then
gosub release
current_zone = zone
endif

exit_rel_test:
return
*****Display Press*****
Press:

serout2 tx,mode,[$24,zone,$50,$2a]
'serout2 tx2,mode,[$24,zone,$50,$2a]

```



```
exit_press:
return
```

```
serout2 tx,mode,[\$24,current_zone,\$52,\$2a]
'serout2 tx2,mode,[\$24,current_zone,\$52,\$2a]
'serout2 tx,mode,["Z ",dec current_zone," Rel ",13,10]
'serout2 tx,mode,[" ",13,10]
```

```
exit_release:
return
```

[illegible][illegible]

1998
 1999
 2000
 2001
 2002
 2003
 2004
 2005
 2006
 2007
 2008
 2009
 2010
 2011
 2012
 2013
 2014
 2015
 2016
 2017
 2018
 2019
 2020
 2021
 2022
 2023
 2024
 2025
 2026
 2027
 2028
 2029
 2030
 2031
 2032
 2033
 2034
 2035
 2036
 2037
 2038
 2039
 2040
 2041
 2042
 2043
 2044
 2045
 2046
 2047
 2048
 2049
 2050
 2051
 2052
 2053
 2054
 2055
 2056
 2057
 2058
 2059
 2060
 2061
 2062
 2063
 2064
 2065
 2066
 2067
 2068
 2069
 2070
 2071
 2072
 2073
 2074
 2075
 2076
 2077
 2078
 2079
 2080
 2081
 2082
 2083
 2084
 2085
 2086
 2087
 2088
 2089
 2090
 2091
 2092
 2093
 2094
 2095
 2096
 2097
 2098
 2099
 2100
 2101
 2102
 2103
 2104
 2105
 2106
 2107
 2108
 2109
 2110
 2111
 2112
 2113
 2114
 2115
 2116
 2117
 2118
 2119
 2120
 2121
 2122
 2123
 2124
 2125
 2126
 2127
 2128
 2129
 2130
 2131
 2132
 2133
 2134
 2135
 2136
 2137
 2138
 2139
 2140
 2141
 2142
 2143
 2144
 2145
 2146
 2147
 2148
 2149
 2150
 2151
 2152
 2153
 2154
 2155
 2156
 2157
 2158
 2159
 2160
 2161
 2162
 2163
 2164
 2165
 2166
 2167
 2168
 2169
 2170
 2171
 2172
 2173
 2174
 2175
 2176
 2177
 2178
 2179
 2180
 2181
 2182
 2183
 2184
 2185
 2186
 2187
 2188
 2189
 2190
 2191
 2192
 2193
 2194
 2195
 2196
 2197
 2198
 2199
 2200
 2201
 2202
 2203
 2204
 2205
 2206
 2207
 2208
 2209
 2210
 2211
 2212
 2213
 2214
 2215
 2216
 2217
 2218
 2219
 2220
 2221
 2222
 2223
 2224
 2225
 2226
 2227
 2228
 2229
 2230
 2231
 2232
 2233
 2234
 2235
 2236
 2237
 2238
 2239
 2240
 2241
 2242
 2243
 2244
 2245
 2246
 2247
 2248
 2249
 2250
 2251
 2252
 2253
 2254
 2255
 2256
 2257
 2258
 2259
 2260
 2261
 2262
 2263
 2264
 2265
 2266
 2267
 2268
 2269
 2270
 2271
 2272
 2273
 2274
 2275
 2276
 2277
 2278
 2279
 2280
 2281
 2282
 2283
 2284
 2285
 2286
 2287
 2288
 2289
 2290
 2291
 2292
 2293
 2294
 2295
 2296
 2297
 2298
 2299
 2300
 2301
 2302
 2303
 2304
 2305
 2306
 2307
 2308
 2309
 2310
 2311
 2312
 2313
 2314
 2315
 2316
 2317
 2318
 2319
 2320
 2321
 2322
 2323
 2324
 2325
 2326
 2327
 2328
 2329
 2330
 2331
 2332
 2333
 2334
 2335
 2336
 2337
 2338
 2339
 2340
 2341
 2342
 2343
 2344
 2345
 2346
 2347
 2348
 2349
 2350
 2351
 2352
 2353
 2354
 2355
 2356
 2357
 2358
 2359
 2360
 2361
 2362
 2363
 2364
 2365
 2366
 2367
 2368
 2369
 2370
 2371
 2372
 2373
 2374
 2375
 2376
 2377
 2378
 2379
 2380
 2381
 2382
 2383
 2384
 2385
 2386
 2387
 2388
 2389
 2390
 2391
 2392
 2393
 2394
 2395
 2396
 2397
 2398
 2399
 2400
 2401
 2402
 2403
 2404
 2405
 2406
 2407
 2408
 2409
 2410
 2411
 2412
 2413
 2414
 2415
 2416
 2417
 2418
 2419
 2420
 2421
 2422
 2423
 2424
 2425
 2426
 2427
 2428
 2429
 2430
 2431
 2432
 2433
 2434
 2435
 2436
 2437
 2438
 2439
 2440
 2441
 2442
 2443
 2444
 2445
 2446
 2447
 2448
 2449
 2450
 2451
 2452

1. *Chlorophyll a* and *Chlorophyll b* were determined by the method of Arar and Collins (1971) using a Shimadzu 1010 spectrophotometer.

Journal of Management Education 30(6)p. 789-804
© The Author(s) 2006